## Contents

1. **Introduction**  
   1.1 Decision Tasks .................................................. 10  
   1.2 Communication .................................................... 11  
   1.3 Failures .............................................................. 11  
   1.4 Timing ............................................................... 12  
   1.4.1 Processes and Protocols ........................................ 12  
   1.5 Chapter Notes ....................................................... 14

2. **Elements of Combinatorial Topology**  
   2.1 The objects and the maps .......................................... 15  
   2.2 Standard constructions ........................................... 18  
   2.3 Chromatic complexes .............................................. 21  
   2.4 Simplicial models in Distributed Computing ...................... 22  
   2.5 Chapter Notes ....................................................... 23  
   2.6 Exercises ............................................................ 23

3. **Manifolds, Impossibility, and Separation**  
   3.1 Manifold Complexes ................................................. 25  
   3.2 Immediate Snapshots ............................................... 26  
   3.3 Manifold Protocols ................................................ 31  
   3.4 Set Agreement ....................................................... 32  
   3.5 Anonymous Protocols .............................................. 36  
   3.6 Weak Symmetry-Breaking .......................................... 37  
   3.7 Anonymous Set Agreement versus Weak Symmetry Breaking .... 38  
   3.8 Chapter Notes ....................................................... 43  
   3.9 Exercises ............................................................ 43

4. **Connectivity**  
   4.1 Consensus and Path-Connectivity .................................. 45
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.2.2</td>
<td>Iterated Immediate Snapshot</td>
<td>110</td>
</tr>
<tr>
<td>7.3</td>
<td>Map Implies Algorithm</td>
<td>111</td>
</tr>
<tr>
<td>7.3.1</td>
<td>Geometric Standard Chromatic Subdivision</td>
<td>111</td>
</tr>
<tr>
<td>7.3.2</td>
<td>Simplicial Approximation</td>
<td>112</td>
</tr>
<tr>
<td>7.3.3</td>
<td>Chromatic Simplicial Approximation</td>
<td>114</td>
</tr>
<tr>
<td>8</td>
<td>Renaming</td>
<td>125</td>
</tr>
<tr>
<td>8.1</td>
<td>Introduction</td>
<td>125</td>
</tr>
<tr>
<td>8.2</td>
<td>A Renaming Algorithm</td>
<td>127</td>
</tr>
<tr>
<td>8.3</td>
<td>Weak Symmetry-Breaking</td>
<td>128</td>
</tr>
<tr>
<td>8.4</td>
<td>The Index Lemma</td>
<td>129</td>
</tr>
<tr>
<td>8.5</td>
<td>The Cone Construction</td>
<td>132</td>
</tr>
<tr>
<td>8.6</td>
<td>Binary Colorings</td>
<td>133</td>
</tr>
<tr>
<td>8.7</td>
<td>A Lower Bound</td>
<td>134</td>
</tr>
<tr>
<td>8.8</td>
<td>Chapter Notes</td>
<td>135</td>
</tr>
<tr>
<td>8.9</td>
<td>Exercises</td>
<td>135</td>
</tr>
</tbody>
</table>
Part I: Undergraduate Course
Chapter 7

Colored Tasks

In Chapter 5, we showed that a colorless task \((I^*, O^*, \Delta^*)\) has a wait-free read-write protocol if and only if there is a continuous map \(f : |I^*| \rightarrow |O^*|\) carried by \(\Delta\). Nevertheless, not all tasks are colorless. For example, the weak symmetry-breaking task discussed in Chapter 3 is not colorless: one process cannot adopt another’s output.

Does this theorem work when tasks are colored? Here is a simple example showing that it does not work. Consider the “hourglass” task whose input and output complexes are shown in Figure 7.1. There are three processes: \(P\), \(Q\), and \(R\), and only one input simplex. The carrier map defining this task is show in schematic form in Figure 7.2 and in tabular form in Figure 7.3. Informally, this task is constructed by taking the standard chromatic subdivision and “pinching” it at the waist to identify (that is, “glue together”) \(P\)’s vertexes on the edges representing its two-process executions.

Note that these complexes satisfy the conditions of Theorem 5.3.6: there is a continuous map \(|I| \rightarrow |O|\) carried by \(\Delta\), shown schematically in Figure 7.4. This map is a homotopy, a continuous deformation of the input complex’s polyhedron into the output complex’s polyhedron.

Nevertheless, despite the fact that the complexes satisfy the conditions of Theorem 5.3.6, there is no wait-free read-write protocol for the Hourglass task. Perhaps the simplest way to see this fact is simply to observe that if we had a read-write protocol for the Hourglass task, then we could solve 2-set agreement.

The protocol is shown in Figure 7.5. The processes share an array announce(), with one entry for each process, initially null. Each process first writes its input value to its own slot in announce(), and then calls the read-write Hourglass protocol. If the Hourglass protocol returns 0, that pro-
cess may be running by itself, so it decides its own input. Otherwise, the processes behave differently. If the Hourglass protocol returns 1 to \( P \), then \( P \) is running concurrently with either \( Q \) or \( R \), so it decides \( \text{announce}(Q) \) or \( \text{announce}(R) \), whichever is not equal to \text{null}. If the Hourglass protocol returns 1 to \( P \) or \( Q \), it decides \( \text{announce}(P) \). If the Hourglass protocol returns 2, the process decides its own input. Figure 7.6 shows the decision values. It is easy to check that in each execution, the processes decide at most two distinct values.

Since there is no wait-free read-write protocol for \( k \)-set agreement, there cannot be a wait-free read-write protocol for the Hourglass task. Why does Theorem ?? fail to hold for colored tasks? One direction still works: it is easy to extend the proof of Theorem ?? to exploit the connectivity of the read-write protocol complex to construct a continuous map \( |\mathcal{I}| \to |\mathcal{P}(\mathcal{I})| \). Composing this map with the decision map yields a continuous map \( |\mathcal{I}| \to |\mathcal{O}| \) carried by \( \Delta \).

The other direction fails. Given a continuous map \( f: |\mathcal{I}| \to |\mathcal{O}| \) carried
Figure 7.2: Carrier Map for Hourglass Task: single-process executions are at the top, executions for \( P \) and \( Q \) on the left (executions for \( P \) and \( R \) are symmetric), and executions for \( Q \) and \( R \) on the right.

<table>
<thead>
<tr>
<th>( \sigma )</th>
<th>( \Delta(\sigma) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( {P} )</td>
<td>( {(P, 0)} )</td>
</tr>
<tr>
<td>( {Q} )</td>
<td>( {(Q, 0)} )</td>
</tr>
<tr>
<td>( {R} )</td>
<td>( {(R, 0)} )</td>
</tr>
<tr>
<td>( {P, Q} )</td>
<td>( {(P, 0), (Q, 1)}, {(P, 1), (Q, 1)}, {(P, 1), (Q, 0)} )</td>
</tr>
<tr>
<td>( {P, R} )</td>
<td>( {(P, 0), (R, 1)}, {(P, 1), (R, 1)}, {(P, 1), (R, 0)} )</td>
</tr>
<tr>
<td>( {Q, R} )</td>
<td>( {(Q, 0), (R, 2)}, {(Q, 2), (R, 2)}, {(Q, 2), (R, 0)} )</td>
</tr>
<tr>
<td>( {P, Q, R} )</td>
<td>( {(P, 0), (Q, 1), (R, 1)}, {(P, 1), (Q, 1), (R, 1)}, {(P, 1), (Q, 0), (R, 2)}, {(P, 1), (Q, 2), (R, 2)}, {(P, 1), (Q, 2), (R, 0)} )</td>
</tr>
</tbody>
</table>

Figure 7.3: The Hourglass Task: tabular specification
by $\Delta$, it is possible to construct a simplicial approximation $\phi : \text{Ch}^N \mathcal{I} \to \mathcal{O}$ carried by $\Delta$, but that simplicial approximation may not be color-preserving. In other words, one process may be assigned another’s output value. Such flexibility is not an issue with colorless tasks, where by definition a process’s inputs and outputs do not depend on its identity. By contrast, for tasks such as weak symmetry-breaking or Hourglass, an output legal for one process may not be legal for another.

### 7.1 Theorem

We are now ready to state our main theorem.

*Theorem* 7.1.1. A task $(\mathcal{I}, \mathcal{O}, \Delta)$ has a wait-free read-write protocol if and only if $\mathcal{I}$ has a chromatic subdivision $\text{Div}(\mathcal{I})$ and a color-preserving simplicial map

$$
\mu : \text{Div} \mathcal{I} \to \mathcal{O}
$$
7.1. **THEOREM**

```c
// Protocol for process P
T[] announce; // initially null

T decide(T input) {
    announce[P] = input;
    int value = Hourglass.decide(P);
    select (value) {
        case 0:
            return announce[P];
        case 1:
            if (announce[Q] != null)
                return announce[Q];
            else
                return announce[R];
    }
}

// Protocol for process Q (R is symmetric)
T decide(T input) {
    announce[Q] = input;
    int value = Hourglass.decide(Q);
    select (value) {
        case 0:
            return announce[Q];
        case 1:
            return announce[P];
        case 2:
            return announce[Q];
    }
}
```

Figure 7.5: Protocol for 2-set agreement using the Hourglass task carried by ∆.

Theorem ?? is depicted schematically in Figure [Diagram]. The figure’s top half shows how the task specification ∆ takes input simplexes to allowed output simplexes. The bottom half shows how µ maps the subdivided input complex to the output complex in a way consistent with ∆.
In Chapter 4 we saw that the (colorless) consensus task has no wait-free solution in read-write memory. Let us relax the consensus task requirements as follows.

**Quasi-Consensus** Each of $P$ and $Q$ is given a binary input. If both have input $v$, then both must decide $v$. If they have mixed inputs, then either they agree, or $Q$ may decide 0 and $P$ may decide 1 (but not vice-versa).

Figure 7.8 shows the input and output complexes for the quasi-consensus task. Does quasi-consensus have a wait-free read-write protocol?

It is easy to see that there is no simplicial map carried by $\Delta$ from the input complex to the output complex. The vertexes of input simplex $\{\langle P, 0 \rangle, \langle Q, 1 \rangle\}$ $\langle P, 0 \rangle$ and $\langle Q, 1 \rangle$, but there is no single output simplex containing both vertexes. Nevertheless, there is a map satisfying the conditions of the theorem from a subdivision of the input complex. If input simplex $\{\langle P, 0 \rangle, \langle Q, 1 \rangle\}$ is subdivided as shown in Figure 7.9, then it can be “folded” around the output complex, allowing input vertexes $\langle P, 0 \rangle$ and $\langle Q, 1 \rangle$ to be mapped to their counterparts in the output complex.

Figure 7.10 shows a simple protocol for quasi-consensus. If $P$ has input 0 and $Q$ has input 1, then this protocol admits three distinct executions: one in which both decide 0, one in which both decide 1, and one in which $Q$ decides 0 and $P$ decides 1. These three executions correspond to the three simplexes in the subdivision of $\{\langle P, 0 \rangle, \langle Q, 1 \rangle\}$, which are carried to $\{\langle P, 0 \rangle, \langle Q, 0 \rangle, \langle P, 1 \rangle, \langle Q, 1 \rangle, \langle P, 1 \rangle, \langle Q, 0 \rangle\}$. 
7.2 Algorithm Implies Map

Assume that we have a read-write protocol $\mathcal{P}(\cdot)$ that solves a task $(\mathcal{I}, \mathcal{O}, \Delta)$. Without loss of generality, we can assume the protocol is in the following normal form: the memory is an array of $n + 1$ unbounded variables, one per process. Initially, a process’s state is its input value. Each process $P_i$ executes a sequence of $r_i$ rounds, where a round consists of writing its state to its variable, and then taking a snapshot of the memory, which becomes $P_i$’s new state. (There may be an arbitrary gap between these steps.)

We will construct a simplicial map $\phi : \text{Ch}^N(\mathcal{I}) \to \mathcal{P}(\mathcal{I})$, where $N = \sum_{i=0}^{n} nr_i$, such that $\phi(\text{Ch}^n(\sigma)) \subset \mathcal{P}(\sigma)$ for $\sigma \in \mathcal{I}$.

To define this map, we will exploit the duality between simplicial complexes and protocol executions by constructing a protocol $\mathcal{Q}(\cdot)$ whose protocol complex $\mathcal{Q}(\sigma)$ is isomorphic to $\text{Ch}(\sigma)$. As this protocol executes, it computes as a “side-effect” the value of $\phi(\vec{q})$ for each vertex $\vec{q}$ of $\mathcal{Q}(\sigma)$.

7.2.1 Immediate Snapshot

We first construct a read-write protocol for the immediate snapshot task whose protocol complex on input simplex $\sigma$ is the standard chromatic subdivision $\text{Ch}(\sigma)$, (This is a chromatic analog to the barycentric agreement protocol of Chapter 5.)

In the immediate snapshot task $(\mathcal{I}, \text{Ch} \mathcal{I}, \text{Ch})$, the processes start on the vertexes of a simplex $\sigma$ in the (colored) complex $\mathcal{I}$, and they halt on the vertex of a simplex in $\text{Ch} \sigma$. Naturally, this task can be repeated to solve the iterated version of the task: $(\mathcal{I}, \text{Ch}^N \mathcal{I}, \text{Ch}^N)$.

The protocol appears in Figure 7.11. Processes share an announce() array, in which each process stores its input value, and level [] array used as follows. Each process starts at level $n+1$. In the loop starting at Line 5, each process lowers its level by 1, and reads the other processes’ levels (Line 7). If the process is at level $\ell$ If there are $i$ processes at the same level or lower, then the process returns their values (Line 9). We claim that this protocol satisfies the invariant that there are always $q+1$ or fewer processes at level $q$ or less. The property clearly holds at the start, when there are no processes. We say that $P_i$ at level $q$ creates a pending descent at Line 7 when it observes there are fewer than $q+1$ processes at level $q$ or less. It completes the pending descent at Line 6 when it updates level $[i]$.

Consider a protocol state where at each level $q$, $0 \leq q \leq n$, there are $k_q$ processes at each level $q$, and $\ell_q$ processes if all pending descents are completed. Following a state change, we denote the new values of these
quantities by $k'_q$ and $\ell'_q$. We will show that the following property is invariant. If all pending descents are completed, then there are $q + 1$ or fewer processes at level $q$ or lower:

$$\sum_{i=0}^{q} \ell_i \leq q + 1.$$  

If this property holds in a state, it cannot be violated by completing a pending descent. What if a process at level $p$ creates a new pending descent? The only value of $\ell'_{q}$ that could change is $q = p - 1$. If a process at level $p$ is about to descend, it must have observed that $\sum_{i=0}^{p} r_i < p + 1$. Completing pending descents at level $p$ and lower can add at most $k_p$ processes to level $p$ and below: $\sum_{i=0}^{p-1} \ell'_i \leq \sum_{i=0}^{p} r_i$. Combining these inequalities yields $\sum_{i=0}^{p-1} \ell'_i \leq p$, satisfying the invariant.

Each process that halts at level $p$ returns the same $p + 1$ values, and the values returned by a process at level $p$ encompass the values returned from level $p - 1$, so the results satisfy the immediate snapshot properties.

### 7.2.2 Iterated Immediate Snapshot

Recall that a read-write protocol $P(\cdot)$ is in normal form if each process $P_i$ executes a sequence of $r_i$ rounds. In each round, each process writes its state to its location in memory, and then takes a memory snapshot, which becomes the process’s new state.

We will construct a simplicial map $\phi : \text{Ch}^n \sigma \to P(\sigma)$. We have seen that applying the immediate snapshot protocol to a complex $\sigma$ yields a chromatic subdivision $\text{Ch} \sigma$. Now imagine that we run the immediate snapshot protocol $N$ times, where each $P_i$’s output for the $i^{th}$ instance of the protocol is $P_i$’s input to the $(i + 1)^{st}$ instance. The resulting complex is the iterated subdivision $\text{Ch}^n \sigma$.

To define this map, we will exploit this duality, where $\text{Ch}^n \sigma$ is both a subdivision of $\sigma$ and a protocol complex for the iterated immediate snapshot protocol. We will construct a full-information iterated immediate snapshot protocol $Q(\cdot)$, where each process’ output from round $i$ is its input to round $i + 1$. For clarity, we refer to writes and snapshots that occur in the read-write protocol $P(\cdot)$ as simulated writes and simulated snapshots.

Starting from input simplex $\sigma$, as the full-information computation unfolds, each process reconstructs a simulated state of the read-write protocol from the simulated states of the other processes. When its simulated state is consistent, the process simulates a snapshot and updates its simulated states. At the end of the computation, the process has computed its local
state for the iterated immediate snapshot protocol, which is a vertex $\vec{s}$ of $\text{Ch}^N(\sigma)$, and its simulated local state for the read-write protocol, which is a vertex $\phi(\vec{s})$ of $\mathcal{P}(\sigma)$.

### 7.3 Map Implies Algorithm

Assume we are given a task $(I, O, \Delta)$, a chromatic subdivision $\text{Div} I$ of the input complex, and a color-preserving map $\phi : \text{Div} I \to O$ carried by $\Delta$. We will show that this task has a wait-free read-write protocol.

The strategy is to show there exists a color-preserving simplicial map $\mu : \text{Ch}^N I \to \text{Div} I$, for some $N > 0$, such that for all $\sigma \in I$, $\mu(\sigma) \subseteq \text{Div} \sigma$. We then construct a protocol as follows. From an input simplex $\sigma$, each process

1. executes an $N$-round iterated immediate snapshot protocol, halting on a vertex $\vec{x}$ of $\text{Ch}^N \sigma$, then
2. computes $\vec{y} = \mu(\vec{x})$, yielding a vertex in $\text{Div} \sigma$, then
3. computes $\vec{z} = \phi(\vec{y})$, yielding an output vertex.

It is easy to check that all processes halt on the vertexes of a single simplex in $\Delta(\sigma)$. Moreover, because all maps are color-preserving, each process halts on an output vertex of matching color.

Because the identity map $\text{Ch}^N I \to \text{Div} I$ is continuous, it has a simplicial approximation $\psi : \text{Ch}^N I \to \text{Div} I$ (Theorem ??). This approximation is a simplicial map carried by $\Delta$. The challenge is to show that this simplicial approximation can be made color-preserving.

Recall that a that a simplex $\sigma = \{\vec{s}_0, \ldots, \vec{s}_n\}$ is chromatic if each vertex is labeled with a distinct color, and a chromatic subdivision $\text{Div} \sigma$ is a subdivision of $\sigma$ where (1) each simplex of the subdivision is chromatic, and (2) for each $\tau \subset \sigma$, each vertex in $\text{Div} \tau$ is labeled with a color from $\tau$.

We must prove the following specialized version of the Simplicial Approximation Theorem:

**Lemma 7.3.1.** If $\sigma$ is a chromatic simplex, and $\text{Div} \sigma$ a chromatic subdivision of $\sigma$, then there exists a color-preserving simplicial map $\mu : \text{Ch}^N \sigma \to \text{Div} \sigma$ such that for all $\tau \subseteq \sigma$, $\mu(\text{Ch}^N \tau) \subseteq \text{Div} \tau$.

### 7.3.1 Geometric Standard Chromatic Subdivision

In Chapter [3] we defined the standard chromatic subdivision $\text{Ch} \sigma$ in a purely combinatorial way, as a simplicial complex. Now we give an equivalent
CHAPTER 7. COLORED TASKS

Definition of Chσ as a subdivision of σ: for each vertex \( \vec{v} \in \text{Chσ} \), we define a point \( |\vec{v}| \in \sigma \).

Assume inductively that for each \( \vec{v} \in \text{ChFace}_i \sigma \), we have defined a corresponding point \( |\vec{v}| \in |\text{Face}_i \sigma| \). Let \( b = \sum_{i=0}^{n} (|\vec{s}_i|/(n + 1)) \) be the barycenter of \( |\sigma| \), where \( n = \text{dim} \sigma \), and \( \delta \) any real value such that \( 0 < \delta < 1/(n + 1) \). For each \( P_i \) and simplex \( \tau \), define

\[ |\langle P_i, \tau \rangle| = (1 + \delta)b - \delta|\vec{s}_i| \]

See Figure ??.

For any value of \( \delta \) such that \( 0 < \delta < 1/(n + 1) \), this definition gives a geometric construction for the chromatic subdivision. We will use this construction for the remainder of this section. Since all simplexes and complexes in this section are geometric, we will not distinguish between an abstract vertex \( \vec{v} \) and the point \( |\vec{v}| \), or an abstract simplex \( \sigma \) and its polyhedron \( |\sigma| \).

Definition 7.3.2. The mesh of a complex is the maximum diameter of any simplex.

Fact 7.3.3. For an \( n \)-simplex \( \sigma \), \( \text{mesh} (\text{Bary} \sigma) \leq \frac{n}{n+1} \text{diam} \sigma \).

Lemma 7.3.4. For an \( n \)-simplex \( \sigma \), there is a constant \( c < 1 \) such that \( \text{mesh} \text{Ch} \sigma \leq c \cdot \text{diam} \sigma \).

Proof. Let \( c = \frac{n}{n+1} \). We argue by induction on \( n \). When \( n \) is zero, the claim is trivial. Let \( \partial \sigma \) be the boundary complex of \( \sigma \), Bary \sigma the barycentric subdivision, and \( \vec{b} \) and \( \vec{b}_i \) the respective barycenters of \( \sigma \) and \( \text{Face}_i \sigma \). Assume inductively that the claim holds for simplexes in \( \text{Ch} \partial \sigma \). From Definition ??, each remaining central vertex \( \vec{x}_i \) has the form \( \vec{x}_i = (1 + \delta)\vec{b} - \delta\vec{s}_i \), which lies on the line joining \( \vec{b} \) to \( \vec{b}_i \). If \( \vec{x} \in \text{ChFace}_i \sigma \), then the edge \( (\vec{x}, \vec{x}_i) \) lies inside the triangle \( (\vec{x}, \vec{b}, \vec{b}_i) \), which lies inside a simplex in Bary \sigma. Since \( \text{mesh} (\text{Bary} \sigma) \leq (n/(n+1)) \text{diam} \sigma \), \( |\vec{x} - \vec{x}_i| \leq (n/(n+1)) \text{diam} \sigma \). Finally, for any central vertex \( \vec{x}_j \), \( |\vec{x}_i - \vec{x}_j| = \delta|\vec{s}_i - \vec{s}_j| \), and the claim follows because \( \delta < 1/(n+1) \).

Lemma ?? implies that by taking sufficiently large \( N \), \( \text{mesh} (\text{Ch}^N I) \) can be made arbitrarily small.

7.3.2 Simplicial Approximation

Definition 7.3.5. An open cover \( \mathbb{U} \) for a complex \( K \) is a finite collection of open sets \( U_0, \ldots, U_k \) such that \( K \subseteq \cup_{i=0}^{k} U_i \).
Specifically, we will be concerned with the open-star cover of $K$ consisting of the open stars of the vertexes in $K$.

**Fact 7.3.6.** If $U_0, \ldots, U_k$ an open cover for $K$, there exists a $\lambda > 0$ (called a Lebesgue number) such that any set of diameter less than $\lambda$ lies in a single $U_i$.

**Lemma 7.3.7.** A set $\{\vec{v}_0, \ldots, \vec{v}_q\}$ is a simplex of $K$ if and only if

$$\bigcap_{i=0}^q \text{Ost}(\vec{v}_i, K) \neq \emptyset.$$  

**Proof.** Left as an exercise. \hfill \Box

**Definition 7.3.8.** Let $\text{Div} \sigma$ be a subdivision of $\sigma$. For any simplex $\tau \in \text{Div} \sigma$, the carrier of $\tau$ in $\text{Div} \sigma$ is the smallest face $\kappa$ of $\sigma$ such that $\tau$ is in $\text{Div} \kappa$. We denote this carrier by $\text{Car}(\tau, \sigma)$, or just $\text{Car} \tau$ when the subdivision is clear from context.

**Definition 7.3.9.** Let $\text{Div}_0 \sigma$ and $\text{Div}_1 \sigma$ be subdivisions of a simplex $\sigma$. A simplicial map $\phi : \text{Div}_0 \sigma \to \text{Div}_1 \sigma$ is carrier-preserving if, for every $\tau \in \text{Div}_1 \sigma$, $\text{Car} \tau = \text{Car} \phi(\tau)$.

The next theorem is a special case of the Simplicial Approximation Theorem (Fact 4.3.1).

**Theorem 7.3.10.** Given an arbitrary subdivision $\text{Div} \sigma$, there is a carrier-preserving simplicial map $\phi : \text{Ch}^N \sigma \to \text{Div} \sigma$ for some $N > 0$.

**Proof.** Let $\lambda$ be the Lebesgue number of the open-star covering of $\text{Div} \sigma$. Pick $N$ large enough that for each vertex $\vec{x} \in \text{Ch}^N \sigma$, the diameter of its star is less than $\lambda$, ensuring the star lies inside the open star of some vertex $\vec{y} \in \text{Div} \sigma$: $\text{St}(\vec{x}, \text{Ch}^N \sigma) \subseteq \text{Ost}(\vec{y}, \text{Div} \sigma)$. For each such $\vec{x}$, define $\phi(\vec{x})$ to be $\vec{y}$.

We must check that $\phi(\cdot)$ is a simplicial map. Let $\{\vec{x}_0, \ldots, \vec{x}_q\}$ be a simplex of $\text{Ch}^N \sigma$. For each $i$, $0 \leq i \leq q$, $\vec{x}_0 \in \text{St}(\vec{x}_i, \text{Ch}^N \sigma)$, so $\vec{x}_0 \in \text{Ost}(\phi(\vec{x}_i), \text{Div} \sigma)$. Because $\vec{x}_0 \in \bigcap_{i=0}^q \text{Ost}(\phi(\vec{x}_i), \text{Div} \sigma)$, the intersection of the open stars is non-empty, so $\{\phi(\vec{x}_0), \ldots, \phi(\vec{x}_q)\}$ is a simplex of $\text{Div} \sigma$ by Lemma 7.3.7. \hfill \Box

This lemma is not quite enough, because our construction does not guarantee that the simplicial map is color-preserving.

Here is how the simplicial map constructed in Lemma 7.3.10 might fail to be color-preserving. Recall that we iterated the standard chromatic subdivision until the simplex diameters were small enough to ensure that the
star of every vertex $\vec{x}$ of $\text{Ch}^N \sigma$ lies inside the open star of some vertex $\vec{y}$ in $\text{Div} \sigma$. The problem is that it might not be possible to to choose $\vec{y}$ so that $\vec{x}$ and $\vec{y}$ have the same color (process ID). The left-hand side of Figure 7.3.2 shows a vertex of $\text{Ch}^N \sigma$, colored by a large star, that happens to lie within a simplex of $\text{Div} \sigma$ whose colors, denoted by a square and circle, do not include a star. The key insight is illustrated on the right-hand side: we can always perturb any such vertex, that is, displace it by an arbitrary small amount, to ensure that it lies in an open star of matching color.

7.3.3 Chromatic Simplicial Approximation

Recall that geometric simplexes “live” in a a Euclidean space of high but finite dimension. Any such space is a metric space, where the distance between points $x$ and $y$ is denoted $|x - y|$. Then $\epsilon$-ball $B_\epsilon(x)$ around a point $x$ is the set of points $y$ such that $|x - y| < \epsilon$, for some $\epsilon > 0$.

We say that a vertex $\vec{v} \in \text{Div} \sigma$ can be perturbed to $\vec{v}'$ if replacing $\vec{v}$ with $\vec{v}'$ in each simplex of $\text{Div} \sigma$ yields a new subdivision $\text{Div}_* \sigma$ isomorphic to $\text{Div} \sigma$. We will see that there is always an $\epsilon > 0$ such that any vertex of a subdivision can be perturbed to any position within $\epsilon$ within its carrier.

First, we show there is room to perturb a vertex.

Lemma 7.3.11. If $\vec{v}$ is a vertex in $\text{Div} \sigma$ with carrier $\kappa$, then there is an open $\epsilon$-ball $B(\vec{v}, \epsilon)$ such that $B(\vec{v}, \epsilon) \cap \kappa$ is contained in $\text{St}(\vec{v}, \kappa)$.

Proof. Because $\text{Div} \tau$ is a manifold, $\text{St}(\vec{v}, \kappa)$ is homeomorphic to a disk. If $\epsilon$ is less than the minimum distance between $\vec{v}$ and $\text{Lk}(\vec{v}, \kappa)$, then $B(\vec{v}, \epsilon) \cap |\kappa|$ is contained in $\text{St}(\vec{v}, \kappa)$. \qed

Definition 7.3.12. A point $y$ is an affine combination of points $x_0, \ldots, x_\ell$ if

$$y = \sum_{i=0}^\ell c_i \cdot x_i$$


maurice: should this def appear in an earlier chapter?

maurice: should this def appear in an earlier chapter?

Definition 7.3.13. A set of points $x_0, \ldots, x_\ell$ is affinely independent if no $x_i$ can be expressed as the affine combination of the others.

Definition 7.3.14. A set of points $x_0, \ldots, x_\ell$ is affinely independent if no $x_i$ can be expressed as the affine combination of the others.
A set of \( k + 1 \) affinely independent vertexes \( \vec{v}_0, \vec{v}_1, \ldots, \vec{v}_k \) determines a hyperplane of dimension \( k \), defined to be the set of points expressible as

\[
\vec{v} = \sum_{i=0}^{k} s_i \cdot \vec{v}_i
\]

where \( \sum_{i=0}^{k} s_i = 1 \). Hyperplanes are closed sets.

**Lemma 7.3.15.** For every vertex \( \vec{v} \) in Div \( \sigma \), there is an \( \epsilon_{\vec{v}} > 0 \) such \( \vec{v} \) can be perturbed to any \( \vec{v}' \) in the open ball \( B(\vec{v}, \epsilon_{\vec{v}}) \).

**Proof.** Let \( \kappa = \text{Car}(\vec{v}, \sigma) \). By Lemma 7.3.11, there exists \( \epsilon > 0 \) such that the open \( \epsilon \)-ball around \( \vec{v} \) in \( |\tau| \) lies in \( \text{Ost}(\vec{v}, \text{Div} \tau) \). Let \( \lambda_0, \ldots, \lambda_q \) be the \((n-1)\)-simplexes of \( \text{Lk}(\vec{v}, \text{Div} \tau) \). For \( 0 \leq i \leq q \), each \( \{vv\} \cup \lambda_i \) is a simplex of \( \text{Div} \tau \), hence \( \vec{v} \) is affinely independent of the vertexes of \( \lambda_i \), meaning that it does not lie in the hyperplane containing \( \lambda_i \). By Lemma 7.3.16, there exists \( \epsilon_i > 0 \) such that every point \( \vec{x} \) of \( \text{Hyper}(\tau) \) within \( \epsilon_i \) of \( \vec{v} \) does not lie on \( \text{Hyper}(\lambda_i) \). Let \( \epsilon = \min(\epsilon, \epsilon_0, \ldots, \epsilon_q) \).

Pick a point \( \vec{x} \) within \( \epsilon \) of \( \vec{v} \) in Div \( \tau \). This point lies within \( \text{Ost}(\vec{v}, \text{Div} \tau) \), and is affinely independent of the vertexes of each \( \lambda_i \). Each \( \mu_i = \{\vec{x}\} \cup \lambda_i \) is thus a simplex, and the set \( \{\mu_i | i \in [q]\} \) is a complex with polyhedron identical to the polyhedron of \( \text{St}(\vec{v}, \text{Div} \tau) \). The desired subdivision \( \text{Div}_* \) is the complex where \( \mu \) replaces \( \text{St}(\vec{v}, \text{Div} \tau) \). \( \square \)

**Lemma 7.3.16.** If \( \text{Div}_* \sigma \) is a perturbation of \( \text{Div} \sigma \) where no vertex is perturbed by more than \( \epsilon \), then \( \text{mesh}(\text{Div}_* \sigma) < 2\epsilon + \text{mesh}(\text{Div} \sigma) \).

**Proof.** The diameter of a simplex is the length of its longest edge. If each vertex in an edge is perturbed by at most \( \epsilon \), then their distance can increase by at most \( 2\epsilon \). \( \square \)

A chromatic simplex is a pair \((\sigma, \text{id})\) where \( \sigma \) is a simplex, and \( \text{id} : \sigma \to [\dim \sigma] \) is a map that assigns a distinct integer value each vertex in \( \sigma \). We call \( \text{id}(\cdot) \) a coloring, and we say that \( \sigma \) is properly colored by \( \text{id}(\cdot) \). For any \( \tau \subseteq \sigma \), \( \text{ids}(\tau) \) denotes the set of colors assigned to vertexes of \( \tau \). Similarly, a chromatic complex is a pair \((\mathcal{C}, \text{id})\) where \( \mathcal{C} \) is a complex, and \( \text{id} : V(\mathcal{C}) \to [\dim \mathcal{C}] \) properly colors each simplex in \( \mathcal{C} \). For brevity, the coloring is usually implicit.

**Definition 7.3.17.** A chromatic subdivision of a simplex \( \sigma \) is a pair \((\text{Div} \sigma, \text{id})\) such that

- \( \text{Div} \sigma \) is a subdivision of \( \sigma \),
• (Div $\sigma, id$) is a chromatic complex, and

• for every $\tau \subseteq \sigma$, $\text{ids}(\text{Div } \tau) = \text{ids}(\tau)$.

All subdivisions considered in this section are chromatic.

**Definition 7.3.18.** An open cover $U$ for a complex $K$ is a finite collection of open sets $U_0, \ldots, U_k$ such that $|K| \subseteq \bigcup_{i=0}^k U_i$.

The open stars of the vertexes in a subdivision Div $\sigma$ form an open cover of $|\sigma|$. This open cover has a natural coloring inherited from the subdivision: $\text{id}($Ost $\vec{v}) = \text{id}(\vec{v})$. We say this coloring is chromatic for Ch $\sigma$ if every simplex in Ch $\sigma$ can be covered by open stars of matching color.

**Definition 7.3.19.** Let Div$_0$ $\sigma$ and Div$_1$ $\sigma$ be chromatic subdivisions. Simplexes $\tau_0$ in Div$_0$ $\sigma$ and $\tau_1$ in Div$_1$ $\sigma$ conflict if their colors do not overlap, but their polyhedrons do: $\text{ids}(\tau_0) \cap \text{ids}(\tau_1) = \emptyset$ but $|\tau_0| \cap |\tau_1| \neq \emptyset$.

**Lemma 7.3.20.** The open-star cover of Div$_0$ $\sigma$ is a chromatic cover for Div$_1$ $\sigma$ if and only if the two subdivisions contain no conflicting simplexes.

**Proof.** Let $\tau_1$ be a simplex in Div$_1$ $\sigma$. The only points in Div$_0$ $\sigma$ that do not lie in an open set colored with an id from $\text{ids}(\tau_1)$ are points in the polyhedron of some simplex $\tau_0 \in$ Div$_0$ $\sigma$ such that $\text{ids}(\tau_0) \cap \text{ids}(\tau_1) = \emptyset$. It follows that $\tau_1$ fails to be covered by open stars colored by $\text{ids}(\tau_1)$ if and only if it conflicts with a $\tau_0 \in$ Div$_0$ $\sigma$.

**Lemma 7.3.21.** Let Div$_0$ $\sigma$ and Div$_1$ $\sigma$ be chromatic subdivisions. There is an $\epsilon$-perturbation Div $\sigma$ of Div$_1$ $\sigma$ for some $\epsilon > 0$, such that the open stars of Div$_0$ $\sigma$ form a chromatic cover for Div $\sigma$.

**Proof.** By Lemma 7.3.20 Div$_0$ $\sigma$ and Div$_1$ $\sigma$ must contain conflicting simplexes $\tau = \{\vec{t}_0, \ldots, \vec{t}_p\}$ and $\rho = \{\vec{r}_0, \ldots, \vec{r}_q\}$. Pick $\tau$ to have minimal dimension. Assume without loss of generality that $\vec{t}_0$ has the carrier of highest dimension among the vertexes of $\tau$. Because $\tau$ and $\rho$ are labeled with $p+q+2$ distinct process IDs, Car $\vec{t}_0$ has dimension at least $p+q+1$. Because $\tau$ and $\rho$ overlap, $\vec{t}_0$ lies in the $(p+q)$-dimensional hyperplane determined by the other vertexes. Since Car $\vec{t}_0$ has dimension at least $p+q+1$, and the hyperplane has dimension $p+q$, any open $\epsilon$-ball around $\vec{t}_0$ includes points not in the hyperplane. We can perturb $\vec{t}_0$ off the hyperplane, yielding a subdivision with one fewer pair of conflicting simplexes. Repeating this process removes all conflicting simplexes.
Lemma 7.3.22. Let Div $\sigma$ be a chromatic subdivision. There exists a carrier and color-preserving simplicial map

$$\mu : \text{Ch}^N \sigma \to \text{Div} \sigma$$

for some $N > 0$.

Proof. Let $\lambda$ be the Lebesgue number of the open cover of $\sigma$ by the open stars of Div $\sigma$. By Lemma 7.3.4, $\text{mesh}(\text{Ch} \sigma) \leq c \text{mesh}(\sigma)$, where $0 < c < 1$. Let $c_* = (1 + c)/2$ (note that $0 < c_* < 1$). By Lemma 7.3.21, Ch $\sigma$ can be perturbed by any $\epsilon$ to an isomorphic subdivision Ch$_* \sigma$ for which the open stars of Div $\sigma$ form a chromatic cover. Pick

$$\epsilon_* < \frac{1 - c}{4} \text{mesh}(\sigma),$$

so by Lemma 7.3.16

$$\text{mesh}(\text{Ch}_* \sigma) \leq 2\epsilon + \text{mesh}(\text{Ch} \sigma)$$

$$\leq \left(\frac{1 - c}{2} + c\right) \text{mesh}(\sigma)$$

$$\leq \frac{c + 1}{2} \text{mesh}(\sigma)$$

$$\leq c_* \text{mesh}(\sigma)$$

Inductively assume we have constructed a subdivision Ch$_*^{N-1} \sigma$ for which the open stars of Div $\sigma$ form a chromatic cover, for which $\text{mesh}(\text{Ch}_*^{N-1} \sigma) \leq c_*^{N-1} \text{mesh}(\sigma)$. We construct another subdivision Ch Ch$_*^{N-1} \sigma$, and perturb it by

$$\epsilon_* < \frac{1 - c}{4} \text{mesh}(\text{Ch}_*^{N-1} \sigma),$$

then the result is a subdivision Ch$_*^N \sigma$ such that

$$\text{mesh}(\text{Ch}_*^N \sigma) \leq c_* \text{mesh}(\text{Ch}_*^{N-1} \sigma)$$

$$\leq c_*^N \text{mesh}(\sigma).$$

Eventually, after enough steps, $\text{mesh}(\text{Ch}_*^N \sigma)$ will be less than the Lebesgue number $\lambda$ for the open star cover of Div $\sigma$, implying that for each vertex $\vec{x} \in \text{Ch}_*^N \sigma$, $\text{St}(\vec{x}, \text{Ch}_*^N \sigma) \subset \text{Ost}(\vec{y}, \text{Div} \sigma)$ for some vertex $\vec{y} \in \text{Div} \sigma$ such that $\text{id}(\vec{x}) = \text{id}(\vec{y})$. Let $\phi(\vec{x}) = \vec{y}$. 
The vertex map $\phi(\cdot)$ is color-preserving by construction. We must check that it is a simplicial map. Let $\{\vec{x}_0, \ldots, \vec{x}_q\}$ be a simplex of $\text{Ch}^N \sigma$. Without loss of generality, we can sort the vertexes so that for $i$, $0 \leq i \leq q$, 
$\{\vec{x}_0, \ldots, \vec{x}_p\} \subset \text{Ost}(\vec{y}_p, \text{Div} \sigma)$, where $\text{id}(vx_p) = \text{id}(\vec{y}_p)$.

The rest of the proof is the same as the proof of Lemma 7.3.10. For each $i$, $0 \leq i \leq q$, $\vec{x}_0 \in \text{St}(\vec{x}_i, \text{Ch}^N \sigma)$, so $\vec{x}_0 \in \text{Ost}(\phi(\vec{x}_i), \text{Div} \sigma)$. Because $\vec{x}_0 \in \bigcap_{i=0}^q \text{Ost}(\phi(\vec{x}_i), \text{Div} \sigma)$, the intersection of the open stars is non-empty, so $\{\phi(\vec{x}_0), \ldots, \phi(\vec{x}_q)\}$ is a simplex of $\text{Div} \sigma$ by Lemma 7.3.7.

$\blacksquare$
Figure 7.8: Input and Output Complexes for 2-Process Quasi-Consensus

Figure 7.9: Subdivided Input and Output Complexes for 2-Process Quasi-Consensus
T[2] announce; // initially null

// code for P
T decide(T input) {
    announce[P] = input;
    if (input == 1)
        return 1;
    else if (announce[Q] != 1)
        return 0
    else
        return 1
}

// code for Q
T decide(T input) {
    announce[P] = input;
    if (input == 0)
        return 0;
    else if (announce[P] != 0)
        return 1
    else
        return 0
}

Figure 7.10: Quasi-Consensus Protocols for P and Q
value\[n+1\] announce; // initially all null
int\[n+1\] level; // initially all n+2
Set[value] immediateSnapshot(int me, value myValue) {
    announce[me] = myValue;
    for (int i = n; n >= 0; n--) {
        level[me] = i;
        int[] view = copy(level);
        Set S = {announce[j] | view[j] <= level[me]};
        if (|S| \geq i)
            return S;
    }
}

Figure 7.11: Immediate Snapshot Protocol

// code for process i
s = input;
for (int r = 0; r < N; r++) {
    s = IS[r].snapshot(s);
}

Figure 7.12: A pure iterated immediate snapshot protocol
// code for process \( i \)
\[
\begin{align*}
M[i] &= \text{input;} \quad // \text{first simulated write} \\
s &= \text{input;} \quad // \text{iterated immediate snapshot state} \\
\textbf{for} \ (\textbf{int} \ r = 0; r < N \ ; r++) \{ \\
\quad (\vec{M}, s) &= \text{IS}[r].\text{snapshot}(s, M); \\
\quad M &= \text{latest}(\vec{M}); \\
\quad W &= \text{writeVec}(M); \\
\quad \textbf{if} \ (W[i] < r_i) \{ \quad // \text{still active?} \\
\qquad \textbf{if} \ (|W| == r) \{ \quad // \text{simulated snapshot complete} \\
\qquad\quad M[i] &= M; \quad // \text{write back simulated snapshot} \\
\qquad\} \\
\} \\
\}
\end{align*}
\]

Figure 7.13: Iterated immediate snapshot protocol that computes simplicial map as side-effect
Figure 7.14: How the construction of Lemma 7.3.10 can fail to be color-preserving (L), and how to fix it (R)
Bibliography


